1. Import Required Libraries

Code:

import re

import openpyxl

import pandas as pd

* **re:** This is Python’s regular expression module, used for string searching and manipulation, such as finding patterns and replacing or removing text.
* **openpyxl:** This library is used for reading and writing Excel files in Python.
* **pandas:** This is a powerful data manipulation library, particularly useful for handling structured data like Excel files, CSVs, etc.

1. Load the Dataset

Code:

file\_path = 'D:/AIUB/Sentiment Analysis Research/Data Sets/Hotel Review Data Table.xlsx'

hotel\_reviews = pd.read\_excel(file\_path, sheet\_name = 0)

* **file\_path:** Specifies the path to the Excel file containing the dataset.
* **pd.read\_excel():** This function loads the Excel file into a Pandas DataFrame. The sheet\_name=0 argument specifies that the first sheet in the Excel file is being loaded.
* **hotel\_reviews:** This variable stores the data in a DataFrame format.

1. Preview the Data

Code:

print(hotel\_reviews.head())

* **hotel\_reviews.head():** This function prints the first 5 rows of the DataFrame to get an overview of the data.

1. Define a Text Cleaning Function

Code:

def clean\_text(text):

# remove html tags

text = re.sub(r'<.\*?>', '', text)

# remove special characters and digits

text = re.sub(r'[^a-zA-Z\s]', '', text)

# convert to lowercase

text = text.lower()

return text

* **def clean\_text(text):** This defines a custom function called clean\_text that takes a text string as input.
* **Remove HTML tags:** The re.sub(r'<.\*?>', '', text) line removes any HTML tags from the text using a regular expression pattern that matches anything between < and >.
* **Remove special characters and digits:** re.sub(r'[^a-zA-Z\s]', '', text) removes everything except alphabetic characters (a-z, A-Z) and spaces.
* **Convert to lowercase:** text.lower() ensures that all text is converted to lowercase, which helps with consistency when processing text.

1. Apply the Cleaning Function

Code:

hotel\_reviews['Cleaned Reviews'] = hotel\_reviews['Review'].apply(clean\_text).

* **hotel\_reviews['Cleaned Reviews']:** This creates a new column called Cleaned Reviews in the DataFrame.
* .**apply(clean\_text):** This applies the clean\_text function to every row in the Review column of the DataFrame, and stores the cleaned text in the new Cleaned Reviews column.

1. Keep Only the Cleaned Reviews

Code:

cleaned\_reviews = hotel\_reviews[['Cleaned Reviews']]

* **cleaned\_reviews:** This new DataFrame contains only the Cleaned Reviews column from the original DataFrame. It drops the other columns, so only the cleaned reviews are retained.

1. Define the Path for the New Excel File

Code:

new\_file\_path = 'D:/AIUB/Sentiment Analysis Research/Data Sets/Cleaned Text Hotel Reviews.xlsx'

* **new\_file\_path:** Specifies the file path where the new cleaned Excel file will be saved.Save the Cleaned Data to a New Excel File

1. Save the Cleaned Data to a New Excel File

Code:

cleaned\_reviews.to\_excel(new\_file\_path, index=False, sheet\_name='Dhaka Regency')

* .**to\_excel():** This saves the cleaned\_reviews DataFrame to a new Excel file at the specified new\_file\_path.
* **index=False:** Prevents Pandas from writing the row numbers (indices) to the Excel file.
* **sheet\_name='Dhaka Regency':** Sets the sheet name of the Excel file as "Dhaka Regency".

1. Print a Confirmation Message

Code:

print(f"Cleaned Text saved to {new\_file\_path}")

* **print(f"..."):** This prints a message to the console, confirming that the cleaned text has been successfully saved to the specified file path.

Summary:

1. Load the hotel review data from an Excel file.
2. Define a function to clean the review text by removing HTML tags, special characters, digits, and converting to lowercase.
3. Apply the cleaning function to the review column.
4. Save the cleaned review data to a new Excel file with a specified sheet name.